Weight saliency in NLDA networks

José R. Dorronsoro, Ana M. González and Carlos Santa Cruz
Department of Computer Engineering and Instituto de Ingeniería del Conocimiento
Universidad Autónoma de Madrid, 28049 Madrid, Spain

Abstract— **In this work we study the architecture selection problem for NLDA networks, a novel feature extraction method proposed by the authors. Although several procedures can be followed for this purpose, we shall concentrate here on the concept of unit saliency, measured in terms of the effect on the NLDA criterion function of variations on the network weights near a minimum. Second order derivative information is needed for this study, whose computation is given for NLDA networks with just one hidden layer. A numerical illustration of the proposed method is also given.**

I. Introduction

Non Linear Discriminant Analysis (NLDA) is a novel method of feature extraction in pattern recognition which has been successfully applied to difficult classification problems with markedly better results than those of Multilayer Perceptrons (MLPs)[4], [12]. The NLDA network architecture is similar to the feedforward one used in Multilayer Perceptrons (MLPs), having one input, one or several hidden layers and one output layer, with sigmoid connections between the input and hidden layers, and linear output connections. The main difference with MLPs is the criterion function employed. Instead of the standard minimum square error, NLDA network training is done by minimizing a Fisher's discriminant analysis like criterion function, such as, for instance, the ratio

$$\mathcal{J}(W) = \frac{|S_W|}{|S_B|},$$

of the determinant of the within class covariance matrix S_W to that of the between class covariance matrix S_B. They are defined as

$$S_B = \sum_1^C N_i (M_i - M)(M_i - M)^t,$$

$$S_W = \sum_{i=1}^C S_i,$$

where N_i, $i = 1, \ldots, C$, is the sample number of elements of the i–th class and M_i that class sample mean, M denotes the total mean and S_i the i–th class covariance matrix. As with MLPs, NLDA network architecture has to be somehow decided upon. Of course, input pattern dimension corresponds to the number of input units and because of the criterion function used, the output layer will have $C-1$ units in a C class problem, as it is customary in Fisher's analysis. The number of hidden layers and of units in each one have still to be settled. This can be done in an empirical fashion, for instance setting before hand the number of hidden layers and then considering for each layer several unit numbers. The corresponding networks are then trained and the final architecture is those of the "best" final network. Several alternatives to this approach have been introduced for MLPs. We shall consider here network "pruning", that is, to start with large networks with many hidden units, whose number is progressively reduced until network performance degrades or there is not a clear indication on whether any further unit has to be taken out. For MLPs, a principled approach to this type of architecture selection can be established from the asymptotic properties of network training. The starting point is to consider network learning as a quasi-maximum "likelihood" estimate [7], [13]. If a large number of sample patterns $\{X_1, X_2, \ldots\}$ is available, and we denote by W_N^* the optimal weight for the N–th sample $\mathcal{S}_N = \{X_1, \ldots, X_N\}$, that is,

$$W_N^* = \arg\min_W \hat{E}(W),$$

where

$$\hat{E}(W) = \frac{1}{N} \sum_1^N ||F(X_i, W) - T^{X_i}||^2,$$

with $F(X, W)$ is the network's transfer function and T^{X_i} an appropriately chosen target vector, it can be then shown that under certain conditions, if these W_N^* converge to a minimum W^* of the distribution-based error function

$$E(W) = \int ||F(X, W) - T^X||^2 f(X) dX,$$

we then have that $\sqrt{N}(W_N^* - W^*)$ converges in distribution to a zero mean normal with covariance matrix $H(W^*)^{-1} I(W^*) H(W^*)^{-1}$. Here denoting the gradient $\nabla_W ||F(X, W^*) - T^X||^2$ as $G(X, W)$, we have

$$H(W^*) = \int \nabla_W G(X, W) f(X) dX,$$

$$I(W^*) = \int G(X, W) G(X, W)^t f(X) dX.$$

I is called the network's Fisher's information matrix. This can be used in two different approaches to network architecture selection. The more general one leads to the Network Information Criteria of Amari [2], [9], which extend to MLP learning the well known Information Criteria of Akaike [1]. An alternative approach uses the preceding convergence result to derive a Wald–like test for weight significance [11], that measures it by the quotient of its square to its variance. According to the previous asymptotic result, this quotient is then for a weight w_i

$$\frac{w_i^2}{(H(W^*)^{-1} I(W^*) H(W^*)^{-1})_{ii}}.$$

This approach lies at the core of network pruning methods such as Optimal Brain Damage (OBD) [8] or Optimal Brain Surgeon (OBS) [6]. Although having a much more heuristic foundation, both methods can be seen as

With partial support from Spain's CICyT, grant TCI 98–965.

providing approximations to the previous ratio. To arrive, for instance, at OBD, the equality $H(W^*) = I(W^*)$, valid under some circumstances, is first used to replace $H(W^*)^{-1}I(W^*)H(W^*)^{-1}$ by $H(W^*)^{-1}$, and then $H(W^*)$ is approximated by the diagonal $\mathrm{diag}(\partial^2\hat{E}/\partial w_i^2)(W_N^*)$ of the Hessian of the sample error function $\hat{E}$. It thus follows that

$$\begin{aligned}(H(W^*)^{-1}I(W^*)H(W^*)^{-1})_{ii} &\simeq (H(W^*)^{-1})_{ii} \\ &\simeq \frac{1}{(\partial^2\hat{E}/\partial w_i^2)(W_N^*)},\end{aligned}$$

and, therefore,

$$\frac{w_i^2}{H(W^*)^{-1}I(W^*)H(W^*)^{-1}}_{ii} \simeq \frac{\partial^2\hat{E}}{\partial w_i^2}(W_N^*)w_i^2.$$

To apply these considerations to NLDA networks we should have to develop first for them the asymptotic theory that lies at the foundation of the preceding MLP discussion, an open question at this moment. Following instead the initial formulation of OBD [8], we will adopt in this work a simpler approach, considering the sensitivity of the NLDA criterion function with respect to network weights. We will start with a Taylor expansion of $\mathcal{J}(W)$ at the vicinity of a minimum W^*, which yields the following formula for the variation $\delta\mathcal{J}$ of the criterion function on a neighborhood of $W^* = (w_i^*)$:

$$\begin{aligned}\delta\mathcal{J} &= \sum_i g_i\delta w_i + \frac{1}{2}\sum_i h_{ii}\delta w_i^2 + \\ &\quad \frac{1}{2}\sum_{i\neq j} h_{ij}\delta w_i\delta w_j + O(||\delta W||^3).\end{aligned}$$

We will disregard cubic and higher order terms. Since we are at a minimum, the coefficients $g_i = \partial\mathcal{J}/\partial w_i(W^*)$ vanish, and we are left with the second order terms, where we introduce a further simplification, looking just at the diagonal terms, which reflect the direct influence of a weight w_i on $\delta\mathcal{J}$. That is, if all the other weights are unchanged, a variation δw_i on w_i results in a variation $\delta\mathcal{J} \simeq h_{ii}\delta w_i^2/2$. This suggests the "weight saliency" value [8]

$$\mathrm{sal}(w_i) = \frac{\partial\mathcal{J}}{\partial w_i}(W^*)w_i^2. \tag{1}$$

We shall use the saliency (1) to prune NLDA network architectures, which makes necessary the computation of the first and second order partials of the criterion function $\mathcal{J}$. NLDA optimal weights are computed in an iterative fashion that combines the classical Fisher eigen–computations [5] for output weights with a numerical, gradient based procedure for the remaining weights. This procedure and also the just mentioned gradient computations are given in [12]; they shall briefly reviewed in the next section, as they are needed for the computation of the second order partials of $\mathcal{J}$. We will give this computation in the third section. The paper will end with a numerical illustration.

II. NLDA networks gradient computation

The weight set W of a general NLDA network can be divided in two groups, which we denote as $W = (W^H, W^O)$. Here W^O denotes the weights of the linear connection between the last hidden and the output layer. W^H denotes the weights connecting the input to the first hidden layer and then all the other hidden layers. Having in mind their update formulae, we will divide the W^H into two weight subsets, $W^H = (W^{H_\ell}, W^{H_p})$, with the W^{H_ℓ} connecting the last two hidden layers, and all other previous connections captured by the W^{H_p}. Notice that in a single hidden layer network (as the ones considered in our illustration) there are no W^{H_p} weights, and therefore $W^H = W^{H_\ell}$.

NLDA weights are computed iteratively, and updates are done in a two step fashion, derived from considering the global criterion function as depending separately on the W^O or the W^H. In other words, assume that the optimal weights $W_{t-1} = (W_{t-1}^O, W_{t-1}^H)$ at the global step $t-1$ have been computed. Then

1. The new W_t^O are obtained by keeping the W_{t-1}^H weights fixed and minimizing $J_t^O(W^O) = \mathcal{J}(W_{t-1}^H, W^O)$. This can be simply done by applying Fisher's classical discriminant analysis having as features the last hidden layer outputs.
2. Once the W_t^O have been computed, we now obtain the W_t^H weights by minimizing $J_t^H(W^H) = \mathcal{J}(W^H, W_t^O)$. But here, in contrast with the simple eigenvalue procedure of Fisher's analysis we can use to obtain the W_t^O, we have to rely now in a purely numerical method, for which we will need to compute the gradient of the criterion $J_t^H(W^H)$. We will show how next.

Given the MLP-like preprocessing of the first hidden layers, W^{H_o} weight gradients can be computed by standard backpropagation once the gradient $\nabla_{W^{H_\ell}} J_t^H(W^H)$ with respect to the W^{H_ℓ} are obtained. The starting point in gradient computations is thus to obtain the partials $\partial J_t^H/\partial w_{kl}^{H_\ell}$, with $w_{kl}^{H_\ell}$ denoting the weight connecting the k–th unit of the preceding layer with the l–th unit of the last hidden layer. For simplicity we will drop in what follows the ℓ subscript and write w_{kl}^H instead of $w_{kl}^{H_\ell}$. These partials are in turn obtained using as auxiliary coordinates the last hidden layer outputs and activations. More precisely we denote as X_{ij} the values at the one-before–the–last hidden layer of the j–th input pattern, $1 \leq j \leq N_i$, of class i, $1 \leq i \leq C$, with N_i the number of sample patterns in class i, and $N = \sum_1^C N_i$ the total number of patterns. If there is just one hidden layer, the X_{ij} are just the network inputs. Assuming D units in this layer, we thus have $X_{ij} = (x_{ij}^1, \ldots, x_{ij}^D)^t$. We also use the notations $a_{ij}^h = \sum_{k=1}^D w_{kh}x_{ij}^k$, $1 \leq h \leq H$, for the activations in unit h of a last hidden layer with H units produced by the X_{ij} and $o_{ij}^h = f(a_{ij}^h)$, $1 \leq h \leq H$, for this layer's outputs (in our illustration f will be the standard sigmoid).

Assuming a two class problem and, hence, a single output, the partial criterion function with respect to the weights W^H reduces to

$$J_t^H(W^H) = \frac{\tilde{s}_W(W^H, W_t^O)}{\tilde{s}_B(W^H, W_t^O)},$$

with $\tilde{s}_W$ and $\tilde{s}_B$ the now scalar valued output covariances. Dropping the t index of the criterion function and using the outputs o_{ij}^h and activations a_{ij}^h as intermediate variables, it follows that

$$\frac{\partial J^H}{\partial w_{kl}^H} = \sum_{i=1}^C\sum_{j=1}^{N_i}\sum_{h=1}^H \frac{\partial J^H}{\partial o_{ij}^h}\frac{\partial o_{ij}^h}{\partial a_{ij}^h}\frac{\partial a_{ij}^h}{\partial w_{kl}^H}$$

$$= \sum_{i=1}^{C}\sum_{j=1}^{N_i}\sum_{h=1}^{H}\frac{\partial J^H}{\partial o_{ij}^h}f'(a_{ij}^h)x_{ij}^k\delta_{hl}$$

$$= \sum_{i=1}^{C}\sum_{j=1}^{N_i}\frac{1}{\tilde{s}_B^2}\left[\tilde{s}_B\frac{\partial\tilde{s}_W}{\partial o_{ij}^l}-\tilde{s}_W\frac{\partial\tilde{s}_B}{\partial o_{ij}^l}\right]f'(a_{ij}^l)x_{ij}^k \quad (2)$$

The scalar valued $\tilde{s}_B$ and $\tilde{s}_W$ can be rewritten using their last hidden layer counterparts S_B and S_W as

$$\tilde{s}_B = \sum_{h=1}^{H}(w_h^O)^2(S_B)_{hh} + 2\sum_{h=1}^{H}\sum_{h'=h+1}^{H}w_h^O w_{h'}^O(S_B)_{hh'},$$

$$\tilde{s}_W = \sum_{h=1}^{H}(w_h^O)^2(S_W)_{hh} + 2\sum_{h=1}^{H}\sum_{h'=h+1}^{H}w_h^O w_{h'}^O(S_W)_{hh'},$$

with w_h^O the weight connecting the h–th unit of the last hidden layer to the network's single output. The hidden layer scatter matrices are now given by

$$S_W = \sum_{i=1}^{C}\sum_{j=1}^{N_i}(O_{ij}-M_i)(O_{ij}-M_i)^t,$$

$$S_B = \sum_{i=1}^{C}N_i(M_i-M)(M_i-M)^t.$$

Here we have used the notation $O_{ij} = (o_{ij}^1,\ldots,o_{ij}^H)^t$, and also $M_i = (m_i^1,\ldots,m_i^H)^t = \frac{1}{N_i}\sum_{j=1}^{N_i}O_{ij}$ for the means M_i of the last hidden layer outputs of the elements of class i, $i=1,\ldots,C$, and $M = (m^1,\ldots,m^H)^t = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N_i}O_{ij}$ for the same outputs' total mean M. It follows from the definitions of M_i and M that

$$\frac{\partial m_c^h}{\partial o_{ij}^l} = \frac{1}{N_i}\delta_{hl}\delta_{ci}, \quad \frac{\partial m^h}{\partial o_{ij}^l} = \frac{1}{N}\delta_{hl}.$$

Since from the S_B and S_W definitions we have

$$\frac{\partial(S_B)_{hh'}}{\partial o_{ij}^l} = \delta_{hl}(m_i^{h'}-m^{h'})+\delta_{h'l}(m_i^h-m^h),$$

$$\frac{\partial(S_W)_{hh'}}{\partial o_{ij}^l} = \delta_{hl}(o_{ij}^{h'}-m_i^{h'})+\delta_{h'l}(o_{ij}^h-m_i^h),$$

it follows that

$$\frac{\partial\tilde{s}_B}{\partial o_{ij}^l} = 2w_l^O\sum_{h=1}^{H}w_h^O(m_i^h-m^h),$$

$$\frac{\partial\tilde{s}_W}{\partial o_{ij}^l} = 2w_l^O\sum_{h=1}^{H}w_h^O(o_{ij}^h-m_i^h). \quad (3)$$

from which the partials $\partial J^H/\partial o_{ij}^l$ and then the $\partial J^H/\partial w_{kl}^H$ can now be easily computed.

III. NLDA networks Hessian computations

In principle, weight saliency, as given by (1) could also be studied on the Fisher weights connecting the last hidden layer to the outputs. However, the corresponding Hessian would have $(C-1)\times H$ rows, but rank $(C-1)\times H-1$, for the Fisher's weights are unique only up to dilations. Thus, although being quite simple to compute, this Hessian will not be considered here. We will thus concentrate on the second partials with respect to the remaining network weights. Again, the MLP–like network architecture allows to use for NLDA nets the second derivative procedures available for MLPs [3], once the second partials have been computed for the W^{H_ℓ} weights, which we do next. The starting point is formula (2), which we write as

$$\frac{\partial J^H}{\partial w_{kl}^H} = \sum_{i=1}^{C}\sum_{j=1}^{N_i}\frac{\partial J}{\partial o_{ij}^l}A_{ij}^{kl}$$

where we will use from now on the notation $A_{pq}^{mn} = f'(a_{pq}^n)x_{pq}^m$. We have now

$$\frac{\partial^2 J^H}{\partial w_{mn}^H\partial w_{kl}^H} = \sum_{i=1}^{C}\sum_{j=1}^{N_i}\frac{\partial^2 J^H}{\partial w_{mn}^H\partial o_{ij}^l}A_{ij}^{kl} + \sum_{i=1}^{C}\sum_{j=1}^{N_i}\frac{\partial J^H}{\partial o_{ij}^l}\frac{\partial A_{ij}^{kl}}{\partial w_{mn}^H}. \quad (4)$$

It is easily seen that in (4)

$$\frac{\partial A_{ij}^{kl}}{\partial w_{mn}^H} = f''(a_{ij}^l)x_{ij}^k x_{ij}^m\delta_{ln},$$

so that the second term can be written in terms of this quantity and the partials $\partial J^H/\partial o_{ij}^l$ already available from gradient computations. We turn our attention to the first sums and, more specifically to the partials $\partial^2 J^H/\partial w_{mn}^H\partial o_{ij}^l$. To compute them, we use again the values o_{pq}^h as intermediate values, and reasoning as in the preceding section, we have

$$\frac{\partial^2 J^H}{\partial w_{mn}^H\partial o_{ij}^l} = \sum_{p=1}^{C}\sum_{q=1}^{N_p}\sum_{h=1}^{H}\frac{\partial^2 J^H}{\partial o_{pq}^h\partial o_{ij}^l}A_{pq}^{mn}\delta_{hn} = \sum_{p=1}^{C}\sum_{q=1}^{N_p}\frac{\partial^2 J^H}{\partial o_{pq}^n\partial o_{ij}^l}A_{pq}^{mn}$$

Using (2) for the second order partials we have

$$\frac{\partial^2 J^H}{\partial o_{pq}^n\partial o_{ij}^l} = \frac{\partial}{\partial o_{pq}^n}\left[\frac{1}{\tilde{s}_B^2}\left(\tilde{s}_B\frac{\partial\tilde{s}_W}{\partial o_{ij}^l}-\tilde{s}_W\frac{\partial\tilde{s}_B}{\partial o_{ij}^l}\right)\right]$$

$$= -\frac{2}{\tilde{s}_B^3}\frac{\partial\tilde{s}_B}{\partial o_{pq}^n}\left(\tilde{s}_B\frac{\partial\tilde{s}_W}{\partial o_{ij}^l}-\tilde{s}_W\frac{\partial\tilde{s}_B}{\partial o_{ij}^l}\right)+\frac{1}{\tilde{s}_B^2}\left(\frac{\partial\tilde{s}_B}{\partial o_{pq}^n}\frac{\partial\tilde{s}_W}{\partial o_{ij}^l}+\tilde{s}_B\frac{\partial^2\tilde{s}_W}{\partial o_{pq}^n\partial o_{ij}^l}\right)-\frac{1}{\tilde{s}_B^2}\left(\frac{\partial\tilde{s}_W}{\partial o_{pq}^n}\frac{\partial\tilde{s}_B}{\partial o_{ij}^l}-\tilde{s}_W\frac{\partial^2\tilde{s}_B}{\partial o_{pq}^n\partial o_{ij}^l}\right)$$

$$= -\frac{1}{\tilde{s}_B^2}\frac{\partial\tilde{s}_B}{\partial o_{pq}^n}\frac{\partial\tilde{s}_W}{\partial o_{ij}^l}+2\frac{\tilde{s}_W}{\tilde{s}_B^3}\frac{\partial\tilde{s}_B}{\partial o_{pq}^n}\frac{\partial\tilde{s}_B}{\partial o_{ij}^l}+\frac{1}{\tilde{s}_B}\frac{\partial^2\tilde{s}_W}{\partial o_{pq}^n\partial o_{ij}^l}-\frac{1}{\tilde{s}_B^2}\frac{\partial\tilde{s}_W}{\partial o_{pq}^n}\frac{\partial\tilde{s}_B}{\partial o_{ij}^l}-\frac{\tilde{s}_W}{\tilde{s}_B^2}\frac{\partial^2\tilde{s}_B}{\partial o_{pq}^n\partial o_{ij}^l}.$$

We will obtain next the values of the last partials using that, from (3), we have

$$\begin{aligned}
\frac{\partial^2 \tilde{s}_B}{\partial o^n_{pq} \partial o^l_{ij}} &= 2 w^O_l \sum_{h=1}^{H} w^O_h \left(\frac{\partial m^h_i}{\partial o^n_{pq}} - \frac{\partial m^h}{\partial o^n_{pq}} \right) \\
&= 2 w^O_l w^O_n \left(\frac{\delta_{ip}}{N_p} - \frac{1}{N} \right), \\
\frac{\partial^2 \tilde{s}_W}{\partial o^n_{pq} \partial o^l_{ij}} &= 2 w^O_l \sum_{h=1}^{H} w^O_h \left(\frac{\partial o^h_{ij}}{\partial o^n_{pq}} - \frac{\partial m^h_i}{\partial o^n_{pq}} \right) \\
&= 2 w^O_l w^O_n (\delta_{ip}\delta_{jq} - \frac{\delta_{ip}}{N_p})
\end{aligned}$$

Therefore, rearranging the preceding partials so that those of second order (the ones that involve new computations) appear first, we have

$$\begin{aligned}
\frac{\partial^2 J^H}{\partial w^H_{mn} \partial o^l_{ij}} &= \frac{1}{\tilde{s}_B} \sum_{p=1}^{C} \sum_{q=1}^{N_p} \frac{\partial^2 \tilde{s}_W}{\partial o^n_{pq} \partial o^l_{ij}} A^{mn}_{pq} - \\
&\quad \frac{\tilde{s}_W}{\tilde{s}^2_B} \sum_{p=1}^{C} \sum_{q=1}^{N_p} \frac{\partial^2 \tilde{s}_B}{\partial o^n_{pq} \partial o^l_{ij}} A^{mn}_{pq} + \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^3_B} \sum_{p=1}^{C} \sum_{q=1}^{N_p} \frac{\partial \tilde{s}_B}{\partial o^n_{pq}} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{mn}_{pq} - \\
&\quad \frac{1}{\tilde{s}^2_B} \sum_{p=1}^{C} \sum_{q=1}^{N_p} \frac{\partial \tilde{s}_B}{\partial o^n_{pq}} \frac{\partial \tilde{s}_W}{\partial o^l_{ij}} A^{mn}_{pq} - \\
&\quad \frac{1}{\tilde{s}^2_B} \sum_{p=1}^{C} \sum_{q=1}^{N_p} \frac{\partial \tilde{s}_W}{\partial o^n_{pq}} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{mn}_{pq} \\
&= \frac{2}{\tilde{s}_B} w^O_l w^O_n \sum_{p=1}^{C} \sum_{q=1}^{N_p} \left(\delta_{ip}\delta_{jq} - \frac{\delta_{ip}}{N_p} \right) A^{mn}_{pq} - \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^2_B} w^O_l w^O_n \sum_{p=1}^{C} \sum_{q=1}^{N_p} \left(\frac{\delta_{ip}}{N_p} - \frac{1}{N} \right) A^{mn}_{pq} + \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^3_B} \sum_{p=1}^{C} \sum_{q=1}^{N_p} \frac{\partial \tilde{s}_B}{\partial o^n_{pq}} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{mn}_{pq} - \\
&\quad \frac{1}{\tilde{s}^2_B} \sum_{p=1}^{C} \sum_{q=1}^{N_p} \frac{\partial \tilde{s}_B}{\partial o^n_{pq}} \frac{\partial \tilde{s}_W}{\partial o^l_{ij}} A^{mn}_{pq} - \\
&\quad \frac{1}{\tilde{s}^2_B} \sum_{p=1}^{C} \sum_{q=1}^{N_p} \frac{\partial \tilde{s}_W}{\partial o^n_{pq}} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{mn}_{pq}.
\end{aligned}$$

Adding the remaining sums,

$$\begin{aligned}
\frac{\partial^2 J^H}{\partial w^H_{mn} \partial w^H_{kl}} &= \sum_{p=1}^{C} \sum_{q=1}^{N_p} \sum_{i=1}^{C} \sum_{j=1}^{N_i} \frac{\partial^2 J^H}{\partial o^n_{pq} \partial o^l_{ij}} A^{mn}_{pq} A^{kl}_{ij} + \\
&\quad \sum_{i=1}^{C} \sum_{j=1}^{N_i} \frac{\partial J^H}{\partial o^l_{ij}} f''(a^l_{ij}) x^m_{ij} x^k_{ij} \delta_{ln} \\
&= \frac{2}{\tilde{s}_B} w^O_l w^O_n \sum_{i,j} \sum_{p,q} \left(\delta_{ip}\delta_{jq} - \frac{\delta_{ip}}{N_p} \right) A^{mn}_{pq} A^{kl}_{ij} - \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^2_B} w^O_l w^O_n \sum_{i,j} \sum_{p,q} \left(\frac{\delta_{ip}}{N_p} - \frac{1}{N} \right) A^{mn}_{pq} A^{kl}_{ij} + \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^3_B} \sum_{i,j} \sum_{p,q} \frac{\partial \tilde{s}_B}{\partial o^n_{pq}} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{mn}_{pq} A^{kl}_{ij} - \\
&\quad \frac{1}{\tilde{s}^2_B} \sum_{i,j} \sum_{p,q} \frac{\partial \tilde{s}_B}{\partial o^n_{pq}} \frac{\partial \tilde{s}_W}{\partial o^l_{ij}} A^{mn}_{pq} A^{kl}_{ij} - \\
&\quad \frac{1}{\tilde{s}^2_B} \sum_{i,j} \sum_{p,q} \frac{\partial \tilde{s}_W}{\partial o^n_{pq}} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{mn}_{pq} A^{kl}_{ij} + \\
&\quad \sum_{i,j} \frac{\partial J^H}{\partial o^l_{ij}} f''(a^l_{ij}) x^m_{ij} x^k_{ij} \delta_{ln} \\
&= \frac{2}{\tilde{s}_B} w^O_l w^O_n \sum_{i,j} A^{kl}_{ij} A^{mn}_{ij} - \\
&\quad \frac{2}{\tilde{s}_B} w^O_l w^O_n \sum_{i=1}^{C} \frac{1}{N_i} \left(\sum_{j=1}^{N_i} A^{kl}_{ij} \right) \left(\sum_{q=1}^{N_i} A^{mn}_{iq} \right) - \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^2_B} w^O_l w^O_n \sum_{i=1}^{C} \frac{1}{N_i} \left(\sum_{j=1}^{N_i} A^{kl}_{ij} \right) \left(\sum_{q=1}^{N_i} A^{mn}_{iq} \right) - \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^2_B} w^O_l w^O_n \frac{1}{N} \left(\sum_{i,j} A^{kl}_{ij} \right) \left(\sum_{p,q} A^{mn}_{pq} \right) + \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^3_B} \left(\sum_{i,j} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{kl}_{ij} \right) \left(\sum_{p,q} \frac{\partial \tilde{s}_B}{\partial o^n_{pq}} A^{mn}_{pq} \right) - \\
&\quad \frac{2}{\tilde{s}^2_B} \left(\sum_{i,j} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{kl}_{ij} \right) \left(\sum_{p,q} \frac{\partial \tilde{s}_W}{\partial o^n_{pq}} A^{mn}_{pq} \right) + \\
&\quad \sum_{i,j} \frac{\partial J^H}{\partial o^l_{ij}} f''(a^l_{ij}) x^m_{ij} x^k_{ij} \delta_{ln} \\
&= \frac{2}{\tilde{s}_B} w^O_l w^O_n \sum_{i,j} A^{kl}_{ij} A^{mn}_{ij} - \\
&\quad \frac{2}{\tilde{s}_B} w^O_l w^O_n \sum_{i=1}^{C} \frac{1}{N_i} \left(\sum_{j=1}^{N_i} A^{kl}_{ij} \right) \left(\sum_{q=1}^{N_i} A^{mn}_{iq} \right) - \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^2_B} w^O_l w^O_n \sum_{i=1}^{C} \frac{1}{N_i} \left(\sum_{j=1}^{N_i} A^{kl}_{ij} \right) \left(\sum_{q=1}^{N_i} A^{mn}_{iq} \right) - \\
&\quad \frac{2\tilde{s}_W}{\tilde{s}^2_B} \frac{1}{N} \left(\sum_{i,j} A^{kl}_{ij} \right) \left(\sum_{p,q} A^{mn}_{pq} \right) - \\
&\quad \frac{2}{\tilde{s}_B} \left(\sum_{i,j} \frac{\partial \tilde{s}_B}{\partial o^l_{ij}} A^{kl}_{ij} \right) \frac{\partial J^H}{\partial w^H_{mn}} + \\
&\quad \sum_{i,j} \frac{\partial J^H}{\partial o^l_{ij}} f''(a^l_{ij}) x^m_{ij} x^k_{ij} \delta_{ln},
\end{aligned}$$

where we have used (2) in the last equality. Notice that at a minimum of J^H, the $\partial J^H / \partial w^H_{mn}$ term will vanish.

As mentioned in the introduction, in this work we shall be interested only on the diagonal terms of the Hessian,

that is, $\partial^2 J^H/\partial w_{kl}^2$, evaluated at minima W^* of J^H. Setting $m = k$ and $n = l$ in the preceding formulae, we have for these points

$$\begin{aligned}\frac{\partial^2 J^H}{\partial w_{kl}^2} &= \frac{2(w_l^O)^2}{\tilde{s}_B}\sum_{i,j}(A_{ij}^{kl})^2 - \\ &\frac{2(w_l^O)^2}{\tilde{s}_B}(1+J^*)\sum_{i=1}^{C}\frac{1}{N_i}\left(\sum_{j=1}^{N_i}A_{ij}^{kl}\right)^2 + \\ &\frac{2(w_l^O)^2}{\tilde{s}_B}\frac{J^*}{N}\left(\sum_{i,j}A_{ij}^{kl}\right)^2 + \\ &\sum_{i,j}\frac{\partial J^H}{\partial o_{ij}^l}f''(a_{ij}^l)(x_{ij}^k)^2,\end{aligned}$$

with $J^* = \tilde{s}_W/\tilde{s}_B$ the minimum value of the target function.

IV. A numerical illustration

We will close this work with an illustration of network pruning by the removal of those weights with the lowest saliencies in a 2 class synthetic classification problem. Both classes are unidimensional, with the first one, C_0, following a $N(0, 0.5)$ distribution and the other one, C_1, being given by a mixture of two gaussians, $N(-2, 0.5)$ and $N(2, 0.5)$. The prior probabilities of these gaussians are 0.5, which is also the prior probability for both classes. As a classification problem, it is an "easy" one, for the mean error probability (MEP) of the optimal Bayes classifier has a rather low value of about 0.48 %. On the other hand, the class distributions are not unimodal, and neither linearly separable. NLDA networks will thus realize feature enhancing rather than feature extraction. Observe that, as it is also the case with MLPs, the outputs of a NLDA network are just a new feature set that can be used to construct an appropriate classifier. A possible way of doing so (and the one used here) is to compute the projections μ_0, μ_1 of the sample means of each class, and use the classifier $\delta_{NLDA} : \{X\} \to \{0, 1\}$ defined as

$$\begin{aligned}\delta(X) &= \delta_{NLDA}(X) \\ &= \text{argmin}_{0,1}\{|F(X, W^*) - \mu_0|, |F(X, W^*) - \mu_1|\},\end{aligned}$$

with F the NLDA network transfer function and W^* the criterion function minimizing weights.

We shall consider simple NLDA networks, with $1 \times H \times 1$ architectures, and a total weight number of $3H$ (H weights connecting the H hidden units to the output, H weights connecting the input to the hidden layer, and one bias parameter for each one of the H hidden units). We shall denote by w_h^H the weights connecting the single input unit to the unit h at the hidden layer and by b_h this unit's bias; w_h^O will denote the weight from this unit h to the single output unit (there are no bias at NLDA output units). It is easily seen that the optimal architecture for this problem just needs 2 hidden units: one hidden unit is not enough for the transformation function used (although it could be if, for instance, $f(x) = x^2$ is used instead of the sigmoidal)

	Hidden unit numbers						
Units	1	2	3	4	5	6	MEP_δ
6	72	71	0	15	129	47	0.48
5	73	0		1	125	43	0.48
4	74			1	128	44	0.49
3	81				135	40	0.48
2	127				133		0.48

TABLE I

Evolution of a saliency based unit removal run. Unit saliencies are shown in the first 5 numerical columns, with empty spaces indicating removed units. Saliency values are rounded to the nearest integer. The last column indicates the MEP value of the resulting classifier.

while with 2 units the MEP of the δ_{NLDA} classifier coincides with the optimal value of 0.48.

In order to apply (1) here to network pruning we have considered *unit* removal rather than just setting a low saliency weight to 0. To do so, we will look at the pairs (w_h^H, b_h^H) just defined and will remove a given hidden unit if the sum of the saliencies of both values are sufficiently small. In other words, in our illustration we will start training networks with a "large" number of hidden units, and to decide then whether or not to remove a hidden unit h depending on the value of its joint saliency, that is

$$\text{sal}(h) = \frac{\partial^2 J^H}{\partial (w_h^H)^2}(W^*)(w_h^{H*})^2 + \frac{\partial^2 J^H}{\partial b_h^2}(W^*)(b_h^*)^2. \quad (5)$$

Another way to discard a hidden unit h could be to apply to the weight w_h^O that connects it to the output one of the tests for feature significance available in Fisher analysis [10]. Although not pursued here, this approach will also be considered in subsequent work.

The initial number of hidden units was 10, and network training was started first at random initial weights. However, at the end of each training session, the unit with the smallest saliency, as given by (5) was deleted, and network training started from the weight values arrived at for the other units. Table I contains the last five steps of the evolution of one such a run. When rows go down, the empty spaces indicate removed units. Notice that for some networks several units could be taken out as having saliencies quite smaller than others. This was not done however, units being removed one by one. The last column of the table shows the MEP values of the δ_{NLDA} classifier after training is finished. All the MEP values are near the Bayes optimum of 0.48%. However, when the lowest saliency unit of a $1 \times 2 \times 1$ network is removed, the MEP value of the resulting $1 \times 1 \times 1$ network shots up (to a 25% value for the run of table I). Similar MEP values were obtained in different runs, although unit saliency magnitudes did differ. In any case, final 2–dimensional hidden features showed a similar structure, with the peaks of the underlying gaussians being projected to a right isosceles rectangle. In the case of table (I), -2 went to the point $(1, 1)$, 2 to $(0, 1)$ and 0 to $(0, 0)$). The final network weights had also similar magnitudes, modulo signs and unit reflections. This seems to indicate a rather robust procedure.

V. Conclusion

We have introduced a pruning procedure for NLDA networks based upon the notion of unit saliencies. They are computed in terms of the Hessian of the network transfer function, for which analytical formulae are given. Although the starting point here for the consideration of saliencies is a second order approximation to the criterion function, saliency can also be seen from the perspective of the statistical testing of the relevance of an estimated parameter. Work is being done in this direction, and also on the use for unit pruning of tests derived from those available in Fisher's analysis.

References

[1] H. Akaike, "A new look at statistical model detection", IEEE Transactions on Automatic Control 19 (1974), 716–723.

[2] S. Amari, N. Murata, "Statistical analysis of regularization constants–from Bayes, MDL and NIC points of view", Proceedings of the IWANN'97, J. Mira, E. Moreno–Díaz, J. Cabestany (eds.), Springer–Verlag, 284–293.

[3] C. Bishop, "Exact calculation of the Hessian matrix for a multilayer perceptron", Neural Computation 4 (1992), 494–501.

[4] J.R. Dorronsoro, A. González, C. Santa Cruz, "Multilayer Perceptrons, Non–Linear Discriminant Analysis and Extreme Sample Discrimination", to appear in Recent Research Developments in Pattern Recognition, Transworld Research, 2000.

[5] R.O. Duda, P.E. Hart, "Pattern classification and scene analysis", Wiley, 1973.

[6] B. Hassibi, D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon", in "Advances in Neural Information Processing Systems" 5, 1993, Morgan Kaufmann, 164–171.

[7] P.J. Huber, "The behavior of maximum likelihood estimates under nonstandard conditions", L.M. Le Cam, J. Neyman (eds.), Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability 1 (1967), 221–233.

[8] Y. Le Cun, J.S. Denker, S.A. Solla, "Optimal Brain Damage", in "Advances in Neural Information Processing Systems" 2, 1990, Morgan Kaufmann, 598–605.

[9] N. Murata, S. Yoshizawa, S. Amari, "Network information criterion–determining the number of hidden units for artificial neural network models", IEEE Transactions on Neural Networks 5 (1994), 965–972.

[10] C.R. Rao, "Linear Statistical Inference and its Applications", Wiley, 1973.

[11] B.D. Ripley, "Pattern Recognition and Neural Networks", Cambridge U. Press, 1996.

[12] C. Santa Cruz, J.R. Dorronsoro, "A nonlinear discriminant algorithm for feature extraction and data classification", IEEE Transactions in Neural Networks 9 (1998), 1370–1376.

[13] H. White, "Learning in artificial networks: a statistical perspective", Neural Computation 1 (1989), 425–464.